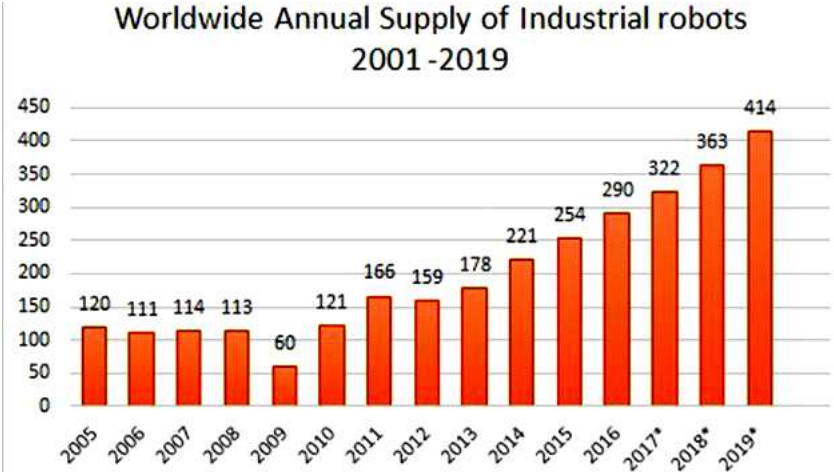
Towards Interactive Verification of Programmable Logic Controllers using Modal Kleene Algebra and KIV

Roland Glück
roland.glueck@dlr.de
Deutsches Zentrum für Luft- und Raumfahrt
Bern, 30th November 2017

Outline

1. Motivation
2. PLC Crash Course
3. Modal Kleene Algebra and Linear Temporal Logic
4. Function Block Diagrams in Modal Kleene Algebra
5. Case Study: Mutual Exclusion
6. Conclusion and Outlook


Worldwide Annual Supply of Industrial robots
2001 -2019
450
400
350
300
250
200
150
100
50
0
120
111
114
113
60
121
166
159
178
221
254
290
322
363
414
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017*
2018*
2019*

robots are:

robots are:

- cost saving

robots are:

- cost saving
- reliable

robots are:

- cost saving
- reliable
- strong

robots are:

- cost saving
- reliable
- strong
- very strong

robots are:

- cost saving
- reliable
- strong
- very strong
- insensitive

robots are:

- cost saving
- reliable
- strong
- very strong
- insensitive
- dangerous

⇒ careful control is indispensable

PLC - Purpose and Function

- Programmable Logic Controllers (PLCs) used for controlling various plants
- robots, pumps, valves, mechanical and automated devices, ...
- PLC works in cyclic way (1 - 150 ms):
  - reads input channels (sensors, switches, internal variables)
  - computes new values
  - writes new values to associated output channels/registers
  - input/output/internal variables

Data Types and Safety

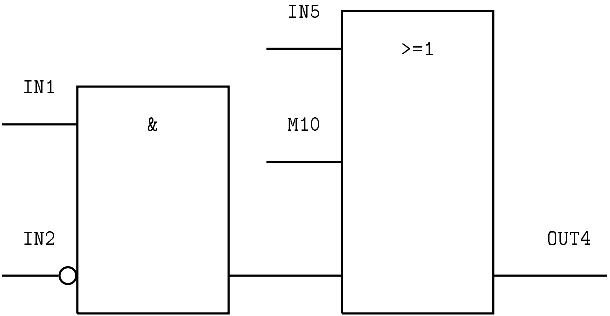
- possible data types: `bool`, `int`, `float`, `date`, ...
- with usual operations (numerical, comparison, ...)
- special part for safety critical operations with reduced instruction set
- from now on only Boolean data and operations

Programming Languages

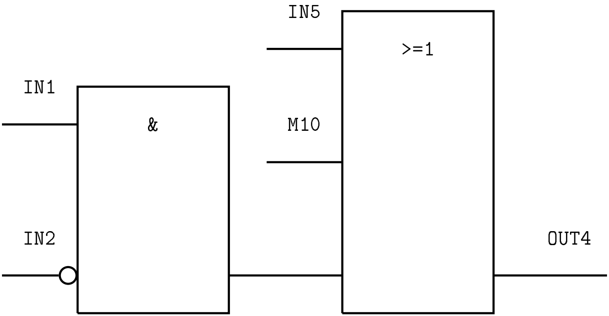
Programming done via:

- Instruction List (IL): assembly-like
- Ladder Diagram (LD): similar to circuit diagrams
- Sequential Function Chart (SFC): inspired by state diagrams
- Structured Text (ST): resembles C syntax
- Function Block Diagram (FBD): see next

AND, OR and Negation in FBD



AND, OR and Negation in FBD

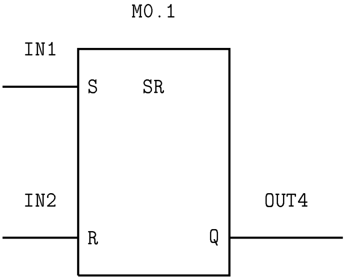


$$\mathtt{OUT4} \equiv (\mathtt{IN1} \land \neg \mathtt{IN2}) \lor \mathtt{IN5} \lor \mathtt{M10}$$

Flip-Flops (Purpose and Function)

- Flip-Flops show dynamic behavior
- two inputs, one marker, one output
- `TRUE`-signal on set input sets output and marker persistently to `TRUE`
- `TRUE`-signal on reset input resets output and marker persistently to `FALSE`
- (until next signal on set/reset input)
- set/reset dominant depending on winner at set/reset conflict
- storing/clearing depending on input signals

Flip-Flops (FBD)



Flip-Flops (Truth Table)

S_n	R_n	Q_{n+1}
TRUE	FALSE	TRUE
FALSE	TRUE	FALSE
FALSE	FALSE	Q_n
TRUE	TRUE	TRUE (set dominant)
TRUE	TRUE	FALSE (reset dominant)

Semirings

Definition

An idempotent *semiring* is a structure $(M, +, 0, \cdot, 1)$ with

- $x + y = y + x$
- $x + (y + z) = (x + y) + z$
- $x + x = x$
- $x + 0 = x$
- $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
- $x \cdot 1 = x = 1 \cdot x$
- $x \cdot 0 = 0 = 0 \cdot x$
- $x(y + z) = xy + xz$ and $(x + y)z = xz + yz$

- $+$ can model choice, $\cdot$ composition
- natural order defined by $x \leq y \;\Leftrightarrow_{df}\; x + y = y$

Semiring examples

- powerset semiring: $(\mathcal{P}(M), \cup, \emptyset, \cap, M)$
- endorelations: $(\mathbf{Rel}(M), \cup, \emptyset, ;, \mathrm{id}_M)$
- monoid semiring: $(\mathcal{P}(M), \cup, \emptyset, \cdot, \{\varepsilon\})$
  for monoid $(M, \cdot, \varepsilon)$ and $\cdot$ lifted to sets
- max-min semiring $(\mathbb{R} \cup \pm\infty, \max, -\infty, \min, \infty)$
- tropical semiring $(\mathbb{R} \cup -\infty, \max, -\infty, +, 0)$
- matrix semiring: $(M^{n\times n}, +, 0^{n\times n}, \cdot, \mathbb{1})$
  defined analogously to conventional linear algebra

Kleene Algebra

Definition

A *Kleene algebra* is a structure $(M, +, 0, \cdot, 1, ^*)$ where $(M, +, 0, \cdot, 1)$ is an idempotent semiring and $^* : M \to M$ has the following properties:

$$1 + xx^* \leq x^* \qquad x + yz \leq z \Rightarrow y^*x \leq z$$
$$1 + x^*x \leq x^* \qquad x + yz \leq y \Rightarrow xz^* \leq y$$

- * models iteration
- $x^* = \sum_{n=0}^{\infty} x^n$ in case of existence
- $a^* = \mu_f = \mu_g$ for $f(x) = 1 + ax$ and $g(x) = 1 + xa$

Kleene Algebra Examples

- endorelations: $(\mathbf{Rel}(M), \cup, \emptyset, ;, \mathrm{id}_M, {}^*)$
- formal languages: $(\mathcal{P}(\Sigma^*), \cup, \emptyset, \cdot, \{\varepsilon\}, {}^*)$
- path algebra: $(\mathcal{P}(\mathbf{path}(A)), \cup, \emptyset, \bowtie, A, A \cup P \cup P \bowtie P \cup P \bowtie P \cup \cdots)$ with

  $$p_1 a \bowtie b p_2 = \begin{cases} p_1 a p_2 & \text{if } a = b \\ \text{undefined} & \text{otherwise,} \end{cases}$$

  lifted to sets
- matrix Kleene algebra: $(M^{n \times n}, +, 0^{n \times n}, \cdot, \mathbb{1}, {}^*)$

  with $\begin{pmatrix} a & b \\ c & d \end{pmatrix}^* = \begin{pmatrix} f^* & f^* b d^* \\ d^* c f^* & d^* + d^* c f^* b d^* \end{pmatrix}$ and $f = a + b d^* c$

Tests

given an idempotent semiring $S = (M, +, 0, \cdot, 1)$ subsets of M can be modeled by tests:

Definition

Given an idempotent semiring $S = (M, +, 0, \cdot, 1)$ an element $p \in M$ is called a *test* if an element $\neg p$ (the *complement* of p) exists with the properties $p + \neg p = 1$ and $p \cdot \neg p = 0 = \neg p \cdot p$.

- set of tests denoted by **test**(S)
- in relational context: subsets of identity
- restriction corresponds to px and xp, resp.

Boxes and Diamonds

(pre)image or (pre I post)condition modeled by diamond/box operators:

Definition

A *modal semiring* is a structure $S = (M, +, 0, \cdot, 1, |\cdot\rangle, \langle\cdot|)$ where $S' = (M, +, 0, \cdot, 1)$ is an idempotent semiring and $|\cdot\rangle$ and $\langle\cdot|$ are functions of the type $M \rightarrow (\textbf{test}(S') \rightarrow \textbf{test}(S'))$ with the properties $|x\rangle p \leq q \Leftrightarrow \neg qxp \leq 0 \Leftrightarrow \langle x|p \leq \neg q$, $|xy\rangle p = |x\rangle|y\rangle p$ and $\langle xy|p = \langle y|\langle x|p$ for all $x \in M$ and $p, q \in S'$.

- $|a\rangle p$: transition into p is possible
- $|a]p =_{df} \neg|a\rangle\neg p$: transition into p is inevitable

Modal Kleene Algebra

putting all together:

Definition

A *modal Kleene algebra* (*MKA* for short) is a structure $(M, +, 0, \cdot, 1, |\cdot\rangle, \langle\cdot|, ^*)$ where $(M, +, 0, \cdot, 1, |\cdot\rangle, \langle\cdot|)$ is a modal semiring and $(M, +, 0, \cdot, 1, ^*)$ is a Kleene algebra.

concrete example: $(\mathbf{Rel}(M), \cup, \emptyset, ;, \mathrm{id}_M, \mathrm{preim}, \mathrm{im}, ^*)$

Modal Kleene Algebra and Linear Temporal Logic

work by Möller, Höfner and Struth (2006):

- model transition system by a general MKA element a
- transforming sets of states into sets of successors
- left total function modeled by $|a\rangle p = |a]p$ for all tests p
- formulae in linear temporal logic (LTL) correspond to expressions in MKA
- LTL formula is valid iff corresponding MKA expression evaluates to 1

Explicit Correspondence

$$
\begin{array}{lcl}
\llbracket \bot \rrbracket & = & 0 \\
\llbracket \neg \psi \rrbracket & = & \neg \llbracket \psi \rrbracket \\
\llbracket \psi_1 \wedge \psi_2 \rrbracket & = & \llbracket \psi_1 \rrbracket \cdot \llbracket \psi_2 \rrbracket \\
\llbracket \psi_1 \vee \psi_2 \rrbracket & = & \llbracket \psi_1 \rrbracket + \llbracket \psi_2 \rrbracket \\
\llbracket \psi_1 \rightarrow \psi_2 \rrbracket & = & \llbracket \psi_1 \rrbracket \rightarrow \llbracket \psi_2 \rrbracket \qquad (p \rightarrow q =_{df} \neg p + q) \\
\llbracket \Box\, \psi \rrbracket & = & |a^*] \llbracket \psi \rrbracket \\
\llbracket \Diamond\, \psi \rrbracket & = & |a^* \rangle \llbracket \psi \rrbracket \\
\llbracket \circ \psi \rrbracket & = & |a \rangle \llbracket \psi \rrbracket \qquad (\text{recall } |a\rangle = |a]) \\
\llbracket \psi_1 \, \mathsf{U} \, \psi_2 \rrbracket & = & |(\llbracket \psi_1 \rrbracket \cdot a)^* \rangle \llbracket \psi_2 \rrbracket
\end{array}
$$

Variables and Overall Behavior

FBDs in MKA:

Variables and Overall Behavior

FBDs in MKA:

- inputs/outputs/internal variables correspond to tests
- for every signal/variable $\mathtt{p}$ introduce two tests $\mathtt{p\_0}$ and $\mathtt{p\_1}$
- indicating a value of $\mathtt{FALSE}$ and $\mathtt{TRUE}$, resp.
- clearly $\neg\mathtt{p\_0} = \mathtt{p\_1}$ and $\neg\mathtt{p\_1} = \mathtt{p\_0}$
- characterize behavior of elementary gates ($\mathtt{OR}$, $\mathtt{AND}$, Flip-Flops, ...)
- elementary gates do not change noninvolved signals/variables
- remember left total functionality
- write overall behavior *a* as product of elementary gates

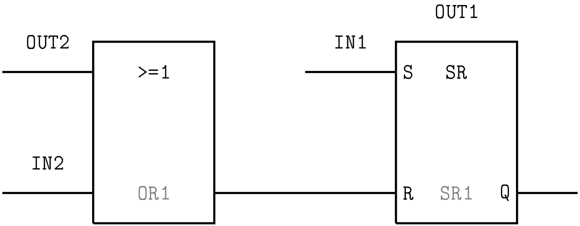
Elementary Gates

- $\texttt{AND}$-gate $\texttt{ANDk}$ with inputs $\texttt{in1},\ \texttt{in2}\ \ldots, \texttt{inn}$:
  - $\texttt{in1\_1} \cdot \texttt{in2\_1} \cdot \cdots \cdot \texttt{inn\_1} \leq |\texttt{andk}\rangle \texttt{andk\_1}$
  - $\texttt{in1\_0} + \texttt{in2\_0} + \cdots + \texttt{inn\_0} \leq |\texttt{andk}\rangle \texttt{andk\_0}$.
- $\texttt{OR}$-gate $\texttt{ORk}$ with inputs $\texttt{in1},\ \texttt{in2}\ \ldots, \texttt{inn}$:
  - $\texttt{in1\_1} + \texttt{in2\_1} + \cdots + \texttt{inn\_1} \leq |\texttt{ork}\rangle \texttt{ork\_1}$
  - $\texttt{in1\_0} \cdot \texttt{in2\_0} \cdot \cdots \cdot \texttt{inn\_0} \leq |\texttt{ork}\rangle \texttt{ork\_0}$.
- negation of $\texttt{sk}$: switch $\texttt{sk\_1}$ and $\texttt{sk\_0}$

Flip-Flops

- set dominant flip-flop RSk with set input s, reset input r, output q and internal marker m :
  - $\texttt{s\_1} + \texttt{m\_1} \cdot \texttt{r\_0} \leq |\texttt{rsk}\rangle \texttt{q\_1}$
  - $\texttt{s\_1} + \texttt{m\_1} \cdot \texttt{r\_0} \leq |\texttt{rsk}\rangle \texttt{m\_1}$
  - $\texttt{s\_0} \cdot \texttt{r\_1} + \texttt{m\_0} \cdot \texttt{s\_0} \leq |\texttt{rsk}\rangle \texttt{q\_0}$
  - $\texttt{s\_0} \cdot \texttt{r\_1} + \texttt{m\_0} \cdot \texttt{s\_0} \leq |\texttt{rsk}\rangle \texttt{m\_0}$

Example Construction (not Complete!)



$$\mathtt{out2\_1} + \mathtt{in2\_1} \leq |\mathtt{or1}\rangle\mathtt{or1\_1}$$

$$\mathtt{out2\_0} \cdot \mathtt{in2\_0} \leq |\mathtt{or1}\rangle\mathtt{or1\_0}$$

$$\mathtt{in1\_0} \leq |\mathtt{or1}\rangle\mathtt{in1\_0}$$

$$\mathtt{in1\_1} \leq |\mathtt{or1}\rangle\mathtt{in1\_1}$$

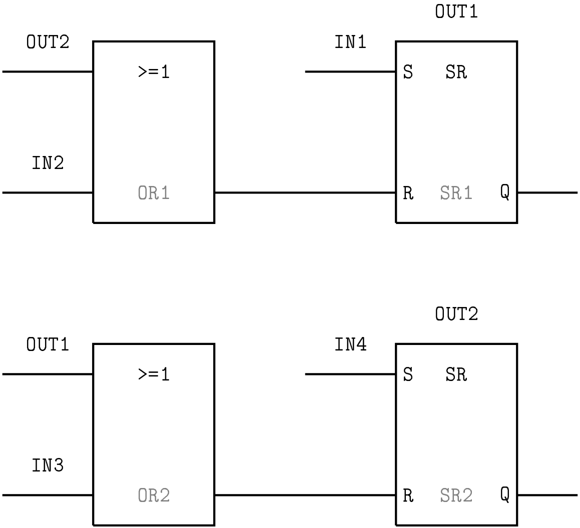
$$|\mathtt{or1}\rangle\mathtt{p} = |\mathtt{or1}]\mathtt{p}$$

$$\mathtt{or1\_1} + \mathtt{out1\_0} \cdot \mathtt{in1\_0} \leq |\mathtt{sr1}\rangle\mathtt{out1\_0}$$

$$\mathtt{in1\_1} \cdot \mathtt{or1\_0} + \mathtt{out1\_1} \cdot \mathtt{or1\_0} \leq |\mathtt{sr1}\rangle\mathtt{out1\_1}$$

$$\mathtt{cycle} = \mathtt{or1} \cdot \mathtt{sr1}$$

Mutual Exclusion



Behavior and Desired Properties

- behavior given by $\texttt{cycle} = \texttt{or1} \cdot \texttt{sr1} \cdot \texttt{or2} \cdot \texttt{sr2}$

Behavior and Desired Properties

- behavior given by $\mathtt{cycle} = \mathtt{or1} \cdot \mathtt{sr1} \cdot \mathtt{or2} \cdot \mathtt{sr2}$
- desired properties in LTL:
  - $\mathtt{out1\_0} \land \mathtt{out2\_0} \rightarrow \Box(\mathtt{out1\_1} \rightarrow \mathtt{out2\_0})$
  - $\mathtt{out1\_0} \land \mathtt{out2\_0} \rightarrow \Box(\mathtt{out2\_1} \rightarrow \mathtt{out1\_0})$

Behavior and Desired Properties

- behavior given by $\mathtt{cycle} = \mathtt{or1} \cdot \mathtt{sr1} \cdot \mathtt{or2} \cdot \mathtt{sr2}$
- desired properties in LTL:
  - $\mathtt{out1\_0} \wedge \mathtt{out2\_0} \rightarrow \Box(\mathtt{out1\_1} \rightarrow \mathtt{out2\_0})$
  - $\mathtt{out1\_0} \wedge \mathtt{out2\_0} \rightarrow \Box(\mathtt{out2\_1} \rightarrow \mathtt{out1\_0})$
- in MKA:
  - $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \rightarrow |\mathtt{cycle}^*](\mathtt{out1\_1} \rightarrow \mathtt{out2\_0}) = 1$
  - $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \rightarrow |\mathtt{cycle}^*](\mathtt{out2\_1} \rightarrow \mathtt{out1\_0}) = 1$

Proof Sketch

to show: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \to |\mathtt{cycle}^*](\mathtt{out1\_1} \to \mathtt{out2\_0}) = 1$

Proof Sketch

to show: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \rightarrow |\mathtt{cycle}^*](\mathtt{out1\_1} \rightarrow \mathtt{out2\_0}) = 1$

proof sketch:

- first: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0}$ is an invariant of `cycle`

Proof Sketch

to show: $\texttt{out1\_0} \cdot \texttt{out2\_0} \to |\texttt{cycle}^*](\texttt{out1\_1} \to \texttt{out2\_0}) = 1$

proof sketch:

- first: $\texttt{out1\_0} \cdot \texttt{out2\_0} + \texttt{out1\_0} \cdot \texttt{out2\_1} + \texttt{out1\_1} \cdot \texttt{out2\_0}$ is an invariant of cycle
- MKA: $\texttt{out1\_0} \cdot \texttt{out2\_0} + \texttt{out1\_0} \cdot \texttt{out2\_1} + \texttt{out1\_1} \cdot \texttt{out2\_0}$ is an invariant of $\texttt{cycle}^*$

Proof Sketch

to show: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \to |\mathtt{cycle}^*](\mathtt{out1\_1} \to \mathtt{out2\_0}) = 1$

proof sketch:

- first: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0}$ is an invariant of $\mathtt{cycle}$
- MKA: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0}$ is an invariant of $\mathtt{cycle}^*$
- MKA: $p \leq q \wedge qx\neg q = 0 \wedge q \leq r \Rightarrow p \to |x]r = 1$

Proof Sketch

to show: $\texttt{out1\_0} \cdot \texttt{out2\_0} \rightarrow |\texttt{cycle}^*](\texttt{out1\_1} \rightarrow \texttt{out2\_0}) = 1$

proof sketch:

- first: $\texttt{out1\_0} \cdot \texttt{out2\_0} + \texttt{out1\_0} \cdot \texttt{out2\_1} + \texttt{out1\_1} \cdot \texttt{out2\_0}$ is an invariant of $\texttt{cycle}$
- MKA: $\texttt{out1\_0} \cdot \texttt{out2\_0} + \texttt{out1\_0} \cdot \texttt{out2\_1} + \texttt{out1\_1} \cdot \texttt{out2\_0}$ is an invariant of $\texttt{cycle}^*$
- MKA: $p \leq q \wedge qx\neg q = 0 \wedge q \leq r \Rightarrow p \rightarrow |x]r = 1$
- finish:
  - $\texttt{out1\_0} \cdot \texttt{out2\_0} \leq \texttt{out1\_0} \cdot \texttt{out2\_0} + \texttt{out1\_0} \cdot \texttt{out2\_1} + \texttt{out1\_1} \cdot \texttt{out2\_0}$
  - $\texttt{out1\_0} \cdot \texttt{out2\_0} + \texttt{out1\_0} \cdot \texttt{out2\_1} + \texttt{out1\_1} \cdot \texttt{out2\_0} \leq \texttt{out1\_1} \rightarrow \texttt{out2\_0}$

DLR

Proof Sketch

to show: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \to [\mathtt{cycle}^*](\mathtt{out1\_1} \to \mathtt{out2\_0}) = 1$

proof sketch:

- first: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0}$ is an invariant of `cycle`
- MKA: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0}$ is an invariant of $\mathtt{cycle}^*$
- MKA: $p \leq q \land qx\neg q = 0 \land q \leq r \Rightarrow p \to [x]r = 1$
- finish:
  - $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \leq \mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0}$
  - $\mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0} \leq \mathtt{out1\_1} \to \mathtt{out2\_0}$
- proof done interactively in KIV

Conclusion

We saw:

Conclusion

We saw:

- Programmable Logic Controllers

Conclusion

We saw:

- Programmable Logic Controllers
- Modal Kleene Algebra

Conclusion

We saw:

- Programmable Logic Controllers
- Modal Kleene Algebra
- Linear Temporal Logic

Conclusion

We saw:

- Programmable Logic Controllers
- Modal Kleene Algebra
- Linear Temporal Logic
- interactive proving with KIV

Conclusion

We saw:

- Programmable Logic Controllers
- Modal Kleene Algebra
- Linear Temporal Logic
- interactive proving with KIV
- and all working together

Outlook

Done:

- formalization of timers
- verification of until-properties

Todo:

- embracing numerical operations
- automated construction of input files
- handling larger systems

References

http://www.ramics-conference.org/
http://www.ens-lyon.fr/LIP/PLUME/RAMiCS17/
http://ramics2015.di.uminho.pt/
https://link.springer.com/chapter/10.1007%2F978-3-319-24704-5_15
https://link.springer.com/chapter/10.1007/11784180_21